

Identificación de cámara fuente a partir de características de textura de fotografías digitales

Diego Alejandro Sarmiento Narvaez

**Proyecto de grado para optar al título de
Ingeniero Físico**

Universidad Tecnológica de Pereira
Facultad de Ingenierías
Ingeniería Física
Pereira, Colombia
2019

Identificación de cámara fuente a partir de características de textura de fotografías digitales

Por:

**Diego Alejandro Sarmiento Narvaez
Cód: 1088351694**

**Proyecto de grado para optar al título de
Ingeniero Físico**

Director:

**M.Sc. Jimmy Alexander Cortés Osorio
Profesor Titular del Departamento de Física
Ingeniero Electricista**

**Universidad Tecnológica de Pereira
Facultad de Ingenierías
Ingeniería Física
Pereira, Colombia
2019**

Índice general

1. Introducción	3
2. Justificación	5
3. Los Objetivos	7
3.1. Objetivo General:	7
3.2. Objetivos Específicos:	7
4. Estado del Arte	8
5. Marco Teórico	11
5.1. Formación de una fotografía digital	11
5.2. Patrones Binarios Locales (LBP)	12
5.3. Cuantificación de Fase Local (LPQ)	13
5.4. Transformada Wavelet	15
6. Metodología	17
6.1. Técnica basada en Patrones Binarios Locales (LBP)	18
6.2. Técnica basada en Patrones Binarios Locales y Cuantificación de Fase Local (LBP-LPQ)	19
6.3. Instancia 1: Evaluación con base de datos Dresden	21
6.4. Instancia 2: Evaluación con galería propia del grupo de investigación	21
7. Resultados	26
7.1. Primera instancia: Evaluación con base de datos Dresden	26
7.2. Segunda instancia: Evaluación con galería propia del grupo de investigación	28
7.2.1. Resultados para el método LBP	28
7.2.2. Resultados para el método LBP-LPQ	35
8. Conclusiones	42
Bibliografía	43

Capítulo 1

Introducción

El desarrollo llevó a la humanidad a una dependencia sustancial de la tecnología, a tal punto que el 67 % de la población mundial cuenta con un dispositivo móvil. En esta época de auge tecnológico, las personas divulga una infinidad de contenido principalmente en forma de fotografías, teniendo la posibilidad de llegar a casi todos los rincones del mundo. Junto a esto, se ha venido presenciando la creación de software más sofisticados para el análisis y modificación de la información entregada por estos dispositivos como Python, Matlab, Photoshop, etc. Diferentes desarrolladores, empresas y usuarios han, desafortunadamente, modificado los principios de funcionamiento de estos programas utilizándolos de forma maliciosa para modificar el contenido de la información.

Las falsificaciones dentro de imágenes digitales, se han convertido en un punto de interés para la informática forense. Estas van desde alteraciones del contenido de la imagen, hasta modificaciones en los meta-datos de la misma. El fácil acceso y el aumento de los software de edición de imágenes han revelado la importancia de la verificación y la autenticidad de las imágenes. Este hecho se vuelve crucial cuando se habla de que estas imágenes son presentadas y vinculadas a un proceso jurídico. Existen una gran cantidad de métodos y técnicas desarrolladas para detectar falsificación de imágenes, tomando como partida el contenido de la misma. Dentro de estos métodos existen algunos que permiten determinar el dispositivo con el que se tomo la fotografía. Lo anterior es posible ya que los procesos de fabricación y los materiales con los que están construidas de las cámaras introducen características estadísticamente únicas a las fotografías tomadas por ellas. Los métodos mencionados consisten en un pre-procesamiento, extracción de características y clasificación.

De acuerdo con lo anterior, se eligieron dos técnicas destacadas de la literatura, basadas en identificación de cámara fuente utilizando las características estadísticas dentro de la fotografía. Estas fueron implementadas y evaluadas con imágenes pertenecientes a la base de datos recomendada y utilizada por los autores en la literatura, y con un conjunto de fotografías tomadas bajo condi-

ciones controladas por el grupo de investigación de Robótica Aplicada en la lineal de Visión por Computador y Aprendizaje de Máquina para determinar su exactitud, entre otras métricas.

Capítulo 2

Justificación

Con el desarrollo de la tecnología desde el siglo XX, la información digital ha tomado gran fuerza, siendo partícipe en todos los campos de la vida influenciando las decisiones y opiniones lo cual revela la importancia de su verificación. En este momento el 67 % de las personas en el mundo poseen acceso a dispositivos móviles de adquisición y propagación de información [1]. En el caso de imágenes digitales, mayormente fotografías, los usuarios pueden modificar el contenido o los datos generales (meta-datos) mediante el uso de software de edición y retoque de imágenes, para posteriormente distribuirlos o presentarlos como testigo silencioso o pruebas sin ser validadas o sin conocer su fuente. Este hecho hace necesario validar la información que llega a diario de forma digital a las personas o cuerpos de justicia. Por esta razón algunas entidades privadas y cuerpos de justicia nacionales promueven la investigación en esta área, ya que no se cuenta con un fácil acceso a los software de verificación o no se conoce un único y verdadero método que permita determinar el año, lugar o dispositivo del que proviene. Debido a esto, en esta investigación se propone analizar los métodos existentes para verificar la fuente de estas imágenes. Las cámaras digitales presentan características únicas las cuales son impregnadas e impuestas por el método de fabricación y los materiales de construcción a las imágenes, haciendo que estas presenten una huella única e inconfundible en su interior. De lo anterior, se pueden derivar diferentes métodos de verificación y obtención de la fuente utilizando variedad de factores propios de las cámaras como lo son: las características de la matriz de filtro de color (CFA), el ruido inducido por el sensor de la cámaras y las características estadísticas dentro de la misma. Estas particularidades estadísticas pueden ser extraídas mediante diferentes variaciones dentro de las imágenes, una de estas es la textura la cual esta directamente relacionada a el tipo de interpolación del color y la capacidad del sensor de la cámara.

A partir de dicho planteamiento, se deriva este proyecto que nace de la necesidad de la Fiscalía General de la Nación seccional Pereira de lograr verificar si las pruebas fotográficas que llegan a sus laboratorios provienen del dispositivo que se afirma. Esta investigación tiene como propósito

principal verificar y validar la procedencia de la información plasmada de forma digital en una fotografía, en lo que se refiere a la cámara fuente, para lo cual es necesario implementar y verificar aquellas técnicas y algoritmos reconocidos en el estado del arte que permitan conocer el dispositivo fuente con el cual se tomó la fotografía. Este proyecto forma parte de todo un conjunto de trabajos liderados por el grupo de Robótica Aplicada, en la línea de Visión por Computador y Aprendizaje de Máquina, de la Universidad Tecnológica de Pereira e incentivados por la Fiscalía General de la Nación, los cuales permiten mediante diferentes métodos o procesos verificar y validar la información contenida en fotografías digitales.

Capítulo 3

Los Objetivos

3.1. Objetivo General:

- Implementar y evaluar un grupo de técnicas que permita identificar la cámara digital de procedencia de una fotografía digital basándose en características de textura entregadas por los operadores conocidos como: Patrones Binarios Locales (LBP) y Cuantificación de Fase local (LPQ).

3.2. Objetivos Específicos:

- Implementar dos técnicas de identificación de cámara fuentes basadas en características estadísticas de textura de una fotografía, la primera utiliza el operador Patrones Binarios Locales (LBP) y la otra técnica es la unión de LBP con la Cuantificación de Fase Local (LPQ).
- Evaluar el grupo de técnicas implementadas para detección de cámara fuente con un conjunto de fotografías digitales.
- Seleccionar la mejor técnica evaluada bajo los criterios de exactitud, robustez y tiempo de procesamiento bajo diferentes parámetros de procesamiento.

Capítulo 4

Estado del Arte

Por la construcción y los algoritmos impuestos por los fabricantes de las diferentes marcas y modelos de las cámaras digitales, las imágenes digitales capturadas tienen patrones o rasgos únicos como: las características de la matriz de filtro de color (CFA), el ruido inducido por el sensor de la cámaras y las características estadísticas dentro de la misma. De acuerdo con lo anterior, dichas particularidades pueden ser usadas para distinguir y rastrear la procedencia de las imágenes partiendo del hecho que estos rasgos representan la huella digital de cada una de las cámaras estudiadas.

La identificación de cámara fuente se ha abordado desde diferentes puntos de vista tomando como punto de partida los rasgos nombrados anteriormente. En Fig.4.1 se presentan algunas de las aproximaciones existentes a este problema.

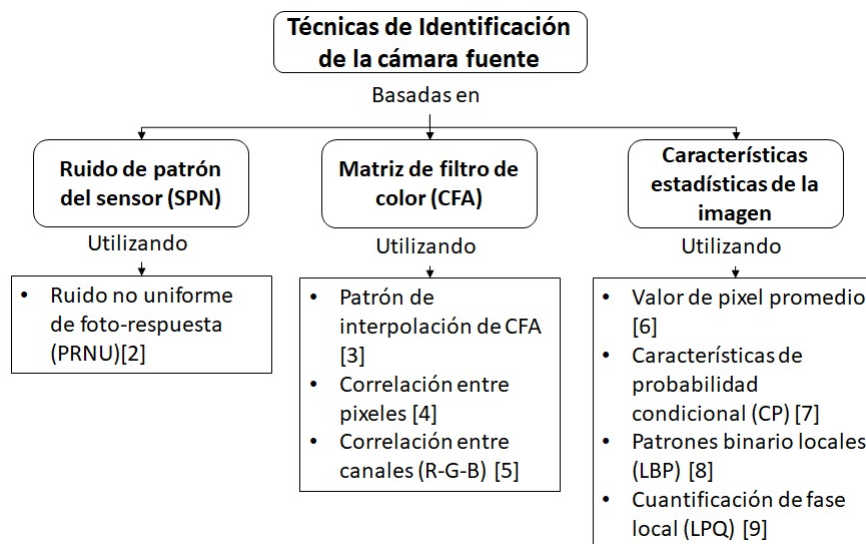


Figura 4.1: Aproximaciones a la identificación de cámara fuente. Fuente: [Autor].

Este proyecto se desarrollara alrededor de las aproximaciones basadas en características estadísticas para la identificación de cámara fuente.

Bo Wang et al. (School of Electronics and Information Dalian University of Technology Dalian, PR China, 2009) presentaron un método aprovechando el principio nombrado inicialmente el cual esta basado en características wavelet para identificar la cámara fuente [10]. B. Wang extrajo características wavelet de orden superior y características de co-ocurrencia de coeficiente wavelet de las imágenes de estudio. Luego aplicó el método de Selección secuencial de características directas (SFFS) para reducir la redundancia y la correlación de características. Finalmente utilizó Máquinas de Soporte Vectorial de múltiples clases (SVM multiclase) como clasificador para identificar las cámaras de origen. El método fue aplicado a seis modelos de cámaras digitales, logrando una precisión promedio del 98 %.

Ainuddin Wahid et al. en [7] (Department of Computer System and Technology, University of Malaya, Malaysia , 2012) explotaron las características de probabilidad condicional (CP) de las imágenes para la identificación de la cámara fuente. Wahib aplicó los CP a los coeficientes de la transformada discreta de coseno (DCT) seleccionada por bloques. Este método alcanzó una exactitud promedio del 99.5 % al ser aplicado a un grupo de 400 imágenes provenientes de 4 modelos de cámaras digitales. El método propuesto presentó una gran robustez cuando las imágenes fueron sometidas a compresión y recorte, entregando un ligero decaimiento del 1.75 % en la exactitud, alcanzando un 97.75 % para el mismo grupo de imágenes.

Guanshuo Xu y Yun Qing Shi en [11] desarrollaron un método en el año 2012 (New Jersey Institute of Technology, Newark, NJ, USA) por el cual extrajeron características estadísticas de la textura de fotografías digitales para determinar la cámara de origen utilizando el principio nombrado inicialmente. En su trabajo extrajeron características de las fotografías mediante el operador de textura denominado “patrones binarios locales” (LBP). Extrajeron características LBP de 59 dimensiones en el dominio espacial de los canales rojo y verde en tres versiones de la misma fotografía, la primera versión estaba constituida por la imagen original, la segunda consistía en la correspondiente matriz de error de predicción de la imagen, y por ultimo se obtuvo la versión de la sub-banda de Wavelets Haar diagonal de nivel 1. Los resultados experimentales arrojaron una exactitud del 98 % para el método propuesto al ser aplicado en un grupo de fotografías tomadas por 18 modelos de cámaras extraídas de la galería Dresden Image Database [12].

Xiaofeng Wang et al. en [13] utilizaron los principios del LBP en el 2016 (Xi'an University of Technology, Xi'an, PR China) para desarrollar un método de identificación de cámara fuente. En este se extrajeron características de la textura de las fotografías, no solo en el dominio espacial si no que también en el dominio de la frecuencia. En su trabajo emplearon el operador LBP acompañado de las características otorgadas por la “cuantificación de fase local” (LPQ), el cual es otro operador de textura pero en el dominio de la frecuencia. Wang desarrolló su método en el espacio

de color HSV en el cual extraen características LBP y LPQ de 59 dimensiones de los canales H y V. El método se aplicó a la imagen original y a la imagen de ruido residual luego, extrajeron características LBP a las matrices de coeficientes representativas de cuatro niveles de alta frecuencia de la transformada de contorno de la imagen de ruido residual. Finalmente, se concatenaron las características extraídas en un solo vector el cual fue llevado a un clasificador basado en máquinas de soporte vectorial (SVM) con función de base radial como función de núcleo para identificar la cámara de origen. La exactitud del método alcanzó un 99.7 % para 11 modelos de cámaras de [12].

Farbod Razzazi y Arash Rashidi en [14] (Faculty of Electrical and Computer Engineering, Islamic Azad University, Tehran, Iran, 2017) adaptaron el método de identificación del hablante conocido como I-Vector para el reconocimiento de cámara fuente, suponiendo que la cámara desempeñaba el mismo papel que el hablante y el contenido de la imagen el mismo que el contenido de la expresión. Razzazi extrajo características de textura de las imágenes para alimentar el sistema de los I-Vectores. Las características fueron extraídas mediante LBP de los canales R y G. Estas se enviaron a un bloque de entrenamiento del modelo mixto Gausiano GMM para calcular los vectores de modelo de fondo universal (UBM) y así calcular el espacio de variabilidad total que conforma el I-Vector de la cámara. Finalmente, los I-Vectores de entrenamiento se introdujeron en una SVM para ser clasificados. El método presentó una exactitud del 99 % al probarse en un grupo de imágenes provenientes de 8 modelos de cámaras tomadas de [12]

Bo Wang et al. en [15] en el 2018 (School of Information and Communication Engineering, Dalian University of Technology, PR China) presentó un método basado en aprendizaje profundo aprovechando las estructuras de las redes neuronales convolucionales (CNN) desarrolladas para el análisis forense en fotografías. Wang utilizó las características LBP para el preprocesamiento de las fotografías para luego ingresarlas a la CNN modificada. El método alcanzó una exactitud del 98.78 % en más de 12 modelos de cámaras de [12] sin ningún otro procedimiento o clasificador.

Capítulo 5

Marco Teórico

5.1. Formación de una fotografía digital

Las cámaras tanto análogas como digitales están compuestas de un par de lentes los cuales son los encargados de redirigir la luz de la escena al cuerpo de la cámara para posteriormente ser plasmada ya sea en un papel fotográfico o en una pantalla LED. En el caso de las cámaras digitales el proceso de adquisición de la escena es un poco complejo, ya que estas cámaras no solo están compuestas de lentes si no que también cuentan con un grupo de filtros, una matriz de filtro de color o CFA, un sensor fotosensible CCD y un procesador [16]. En Fig. 5.1 se observa la estructura básica o pipeline del cuerpo de una cámara digital [17].

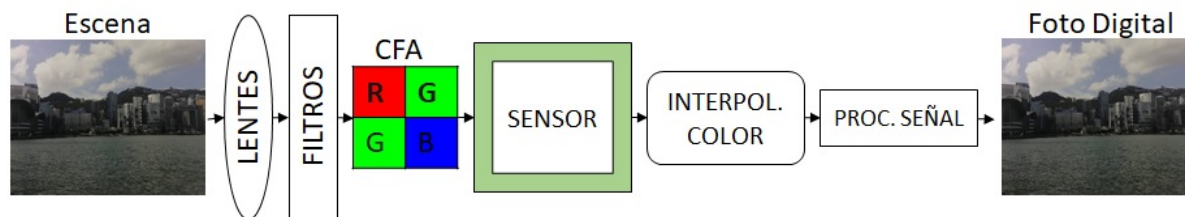


Figura 5.1: Proceso de creación de una fotografía digital. Fuente: [Autor].

Los fabricantes de estas cámaras mantienen bajo confidencialidad gran parte del procesamiento además del tipo de CFA, en la mayoría de casos los procesos son similares entre una casa fabricante y otra [18].

El primer paso de la generación de una fotografía en una cámara es que la luz de la escena pase a través de los lentes los cuales regulan y redirigen la luz. Después, esta pasa a través de unos filtros los cuales mejoran la calidad de la imagen y evitar que radiación no deseada llegue a los demás sistemas [18]. Posteriormente ser llevada a lo que se conoce como CFA la cual esta compuesta por un mosaico de minúsculos filtros de color colocados sobre los píxeles del sensor CCD para capturar la información del color de la escena. Los filtros presentan una configuración determinada por las

casas fabricantes y la configuración más común es conocida como arreglo Bayer el cual se ilustra en Fig. 5.2. Este arreglo de filtros es de vital importancia en el pipeline ya que el sensor no es capaz de reconocer la longitud de onda que llega a él sino la intensidad con la que arriba, por lo tanto no pueden identificar esta información.

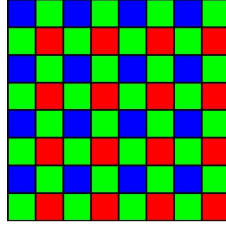


Figura 5.2: Arreglo Bayer. Fuente: [Autor].

Una vez el sensor registra la luz la envía al procesador de la cámara, donde este se encarga de interpolar la información del sensor y la suministrada por la CFA para recrear los colores y la escena tomada. Por último, esta información se almacena en la memoria de la cámara tal cual sale del procesador o de forma comprimida si se desea.

5.2. Patrones Binarios Locales (LBP)

Los Patrones Binarios Locales (LBP) fueron introducidos por primera vez por Ojala et al. en [8] para caracterizar información en píxeles de imágenes con textura. El operador LBP tiene un alto poder discriminatorio y un bajo costo computación, haciéndolo más simple pero muy eficiente. Este hecho aumentó su popularidad y uso en diversas aplicaciones en el procesamiento digital de imágenes.

Ojala [8] inicialmente propuso este operador en escala de grises, el cual etiquetaba píxeles al recorrer la imagen con un umbral circular simétrico de radio R y número de vecinos P , el resultado de este etiquetado se considera un valor binario. Matemáticamente se puede expresar como se muestra en (5.1).

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) * 2^i \quad (5.1)$$

En este documento se estableció $R = 1$ y $P = 8$. Donde g_i y g_c son el valor de intensidad del píxel vecino y el píxel central respectivamente, s esta definida por (5.2).

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (5.2)$$

De (5.1) y (5.2) se calcula la diferencia de intensidades entre los píxeles vecinos y el pixel central dentro del umbral, esta diferencia es cuantificada de forma binaria produciendo así patrones binarios locales. Luego, el histograma de la cuantificación es usado como descriptor de textura de la imagen.

Al rotar la imagen el pixel central no cambia de posición pero sus vecinos si, este hecho llevo a los autores a tomar medidas he introducir el concepto de patrones binarios locales “invariantes bajo rotación”, la decisión para solucionar este inconveniente fue rotar $P - 1$ veces la posición de los vecinos dentro del umbral y escoger el menor valor de cuantificación generados por las rotaciones. Ademas, se introducen dos nuevos conceptos: patrones binario locales “uniformes” y “no uniformes”. Se considera uniformidad dentro del umbral si, el número de cambios binarios es igual a dos o menos (ver Fig. 5.3). Los LBP uniformes están distribuidos sobre casi todas las 2^P posiciones el histograma, por otro lado, los no uniformes se fusionan en una posición del histograma, lo que lleva de 256 casillas a 59 si $P = 8$.

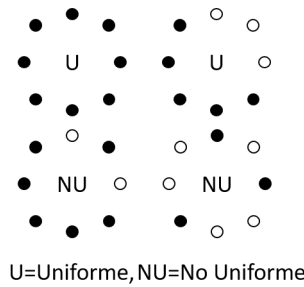


Figura 5.3: LBP “uniformes” y “no uniformes”. Fuente: [Autor].

5.3. Cuantificación de Fase Local (LPQ)

El operador Cuantificación de Fase Local o LPQ fue propuesto por V. Ojansivu y J. Heikkilä en [9] principalmente como descriptor y clasificador de textura insensible al desenfoque.

En el procesamiento digital de imágenes, el modelo de desenfoque espacialmente invariante de una imagen $f(x)$ se puede expresar como la convolución entre la imagen y la función de punto de dispersión (PSF) (5.3).

$$g(x) = (f * h)(x) \quad (5.3)$$

Donde $g(x)$ es la imagen difuminada, $h(x)$ representa la función de punto de dispersión, $*$ la convolución 2-D y x el vector de coordenadas $[x, y]^T$. La imagen borrosa también puede definirse en el dominio de la frecuencia como $G(u)$, la cual es la multiplicación de las transformadas discretas de Fourier (DFT) de la imagen original $f(x)$ y la función de punto de dispersión $h(x)$ (5.4), donde

$$\mathbf{u} = [u, v]^T.$$

$$G(\mathbf{u}) = F(\mathbf{u}) \cdot H(\mathbf{u}) \quad (5.4)$$

Si consideramos la fase y la magnitud de (5.4) por separado obtenemos (5.5) y (5.6).

$$\angle G(\mathbf{u}) = \angle F(\mathbf{u}) + \angle H(\mathbf{u}) \quad (5.5)$$

$$|G(\mathbf{u})| = |F(\mathbf{u})| \cdot |H(\mathbf{u})| \quad (5.6)$$

Si la PSF es simétrica, su DFT presentara un valor real y un angulo de fase igual a 0 si $H(\mathbf{u}) \geq 0$ ó π si $H(\mathbf{u}) < 0$ de tan forma que $\angle G(\mathbf{u}) = \angle F(\mathbf{u})$ si $H(\mathbf{u}) \geq 0$. Lo anterior hace que las frecuencias bajas de H sean positivas, lo que causa que $\angle F(\mathbf{u})$ sea 0 brindando así la propiedad de insensibilidad al borrón. Bajo esta propiedad se basa el LPQ.

La Cuantificación de Fase Local es un método el cual cuantifica el espectro de la fase de Fourier, la cual proporciona un descripción insensible al desenfoque de una imagen. Esta información se extrae utilizando la DFT 2-D con un umbral rectangular de tamaño M en una vecindad N_x en cada posición de pixel x de la imagen original como se indica en (5.7).

$$F(\mathbf{u}, x) = \sum_{y \in N_x} f(x - y) e^{-j2\pi \mathbf{u}^T y} = \mathbf{w}_u^T \mathbf{f}_x \quad (5.7)$$

Donde \mathbf{w}_u es el vector básico de la DFT 2-D de la frecuencia \mathbf{u} .

El operador LPQ solo considera cuatro coeficientes complejos, correspondientes a las frecuencias 2-D $\mathbf{u}_1 = [a, 0]^T$, $\mathbf{u}_2 = [0, a]^T$, $\mathbf{u}_3 = [a, a]^T$ y $\mathbf{u}_4 = [a, -a]^T$, donde a es una frecuencia escalar que cumple con la condición de $H(\mathbf{u}) \geq 0$. entonces para cada pixel x tenemos (5.8) y (5.9).

$$\mathbf{F}_x^c = [F(\mathbf{u}_1, x), F(\mathbf{u}_2, x), F(\mathbf{u}_3, x), F(\mathbf{u}_4, x)] \quad (5.8)$$

$$\mathbf{F}_x = [\text{Re}\{\mathbf{F}_x^c\}, \text{Im}\{\mathbf{F}_x^c\}]^T \quad (5.9)$$

Donde $\text{Re}\{\cdot\}$ y $\text{Im}\{\cdot\}$ representan la parte real e imaginaria del numero complejo. De forma matricial (5.9) quedaría como en (5.10).

$$\mathbf{F}_x = \mathbf{W} \mathbf{f}_x \quad (5.10)$$

$$\mathbf{W} = [\text{Re}\{\mathbf{w}_{u_1}, \mathbf{w}_{u_2}, \mathbf{w}_{u_3}, \mathbf{w}_{u_4}\}, \text{Im}\{\mathbf{w}_{u_1}, \mathbf{w}_{u_2}, \mathbf{w}_{u_3}, \mathbf{w}_{u_4}\}] \quad (5.11)$$

Antes de la cuantificación se define la transformación de blanqueamiento presentada en (5.12).

$$G_x = V^T F_x \quad (5.12)$$

Donde V es una matriz ortonormal derivada de la descomposición del valor singular (SVD) de la matriz de covarianza D del vector de transformación F_x .

5.4. Transformada Wavelet

Las Wavelets dan como resultado una potente representación de información en el dominio temporal y en frecuencia simultáneamente. La descomposición Wavelet tiene una gran potencia al analizar señales e imágenes con discontinuidades o puntas en su interior por lo que realza la importancia y utilidad en el procesamiento digital de imágenes. Entre sus aplicaciones se encuentra la compresión y filtrado de imágenes [19].

Si se habla de imágenes digitales las Wavelets descomponen la información de la imagen en diferentes frecuencias que entregan niveles diferentes de detalle. Como se observa en la Fig.5.4b el componente de la esquina superior izquierda es una copia exacta de la imagen original Fig.5.4a pero en menor calidad, a esta componente se le conoce como componente de baja frecuencia L . El resto de representaciones se les conoce como componentes de alta frecuencia, estos contienen como su nombre indica la información de alta frecuencia. Los componentes que contienen las frecuencias altas son llamados H , V y D , representando los cambios locales en las direcciones horizontal, vertical y diagonal respectivamente [18].

Para crear los niveles de descomposición las Wavelets aplican una función recursiva como se observa en Fig.5.4d, en Fig.5.4c se observa la descomposición de cuarto nivel de la imagen.

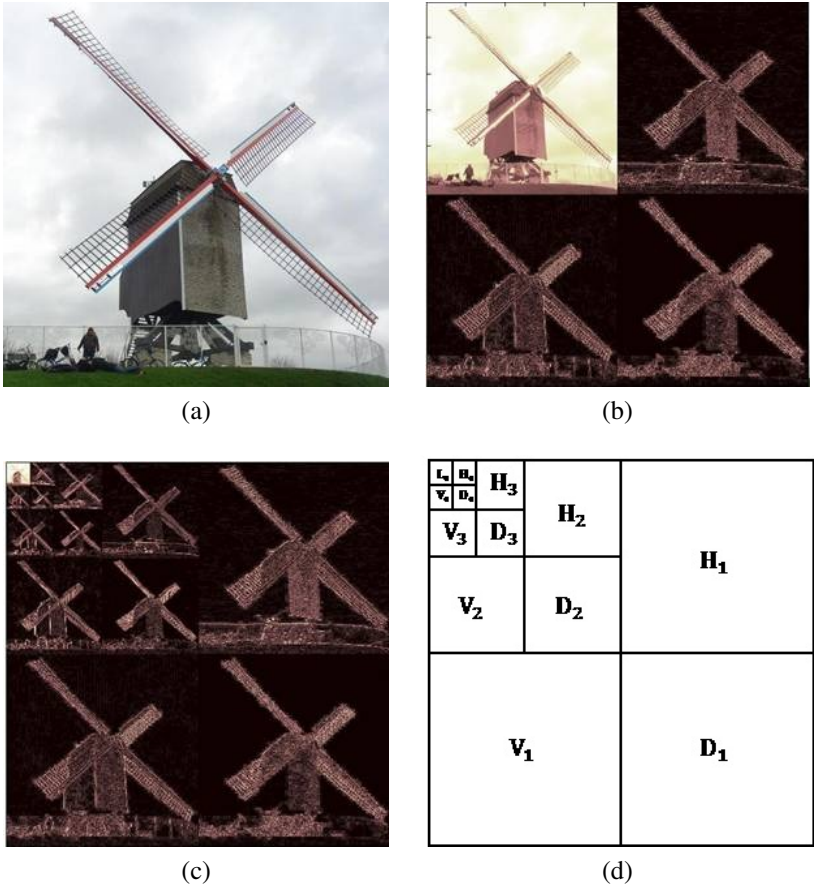


Figura 5.4: Descomposición wavelet de cuarto nivel de una imagen [18].

Capítulo 6

Metodología

Para la realización de este proyecto de investigación aplicada, se plantearon y desarrollaron unas etapas con el fin de cumplir los objetivos propuestos.

La primera etapa la cual se desarrolló para cumplir el primer objetivo de implementar las técnicas, consistió en analizar artículos basados en identificación de cámara fuente a partir de fotografías digitales de las bases de datos de la IEEE, Elsevier, Neurocomputing y Springer para una mayor comprensión del universo de técnicas existentes y los tipos de preprocesamiento de estas [16, 20, 21, 22]. Con las investigaciones encontradas se creó una base de datos la cual contenía la información general de los trabajos.

Con la base de datos creada, se procedió a escoger dentro de ella aquellos trabajos que utilizaron técnicas basadas en características estadísticas, que fueran más reconocidos, que contaran con la información necesaria para su implementación y en lo posible que su año de publicación fuese el más reciente ya que con esto último se garantizó que las técnicas fueran lo más recientes posibles. Luego de hacer este primer filtrado a la base de datos bibliográficos, se eligieron las técnicas cuyo método se fundó específicamente en identificación bajo características de textura de las fotografías [14, 15, 11, 13]. Estos se basaron en extracción de características de textura mediante: Patrones Binarios Locales (LBP) [8] y Cuantificación de Fase Local (LPQ) [9]. Los anteriores se estudiaron a profundidad y de estos se eligieron los trabajos de Guanshuo Xu [11] y Xiaofeng Wang [13] ya que contaban con la información necesaria para sus implementaciones y sus resultados fueron satisfactorios. Estos se usaron como referencia primaria para el desarrollo de algoritmos en lenguaje *m* en el sistema de análisis numérico Matlab, con los cuales se extrajeron las características propias para cada técnica (cuadro 6.1) y se obtuvieron las representaciones de las cámaras.

Técnica	Características Propias
LBP [11]	Histograma en el dominio espacial
LBP-LPQ [13]	Histograma en el dominio espacial y en la frecuencia

Cuadro 6.1: Listado de técnicas y sus respectivas características propias.

Los algoritmos mencionados anteriormente se desarrollaron y ejecutaron mediante la versión 2017b de Matlab, en un equipo Dell de 64bits, 8GB de RAM con Windows 10 y procesador Intel® Xeon. Para obtener las componentes significativas de las técnicas presentes en el cuadro 6.1 se realizaron los procesos nombrados a continuación:

6.1. Técnica basada en Patrones Binarios Locales (LBP)

La técnica consistió en extraer características LBP de 59-D de las fotografías en el modelo de color RGB de los canales rojo y verde de las versiones originales, de las imágenes de error de predicción (PE) y de las sub-bandas HH de la transformada Wavelet Haar de primer nivel [11]. El canal azul se despreció por el hecho de que este posee la misma información que el canal rojo al tener el mismo procesamiento dentro de las cámaras en términos de la matriz bayer [23]. El diagrama de flujo del método se presenta en la Fig. 6.1.

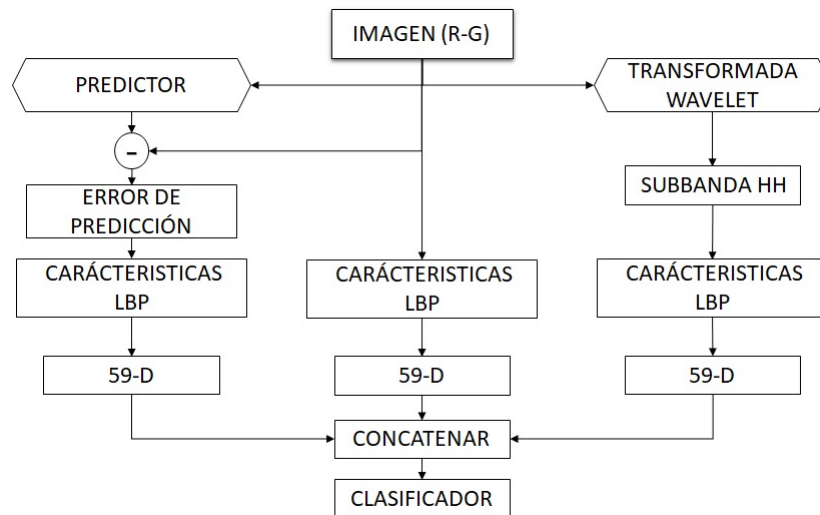


Figura 6.1: Diagrama de flujo método 1. Fuente: [Autor].

La imagen PE se obtuvo restando la imagen predicha de la imagen original. Para calcular la imagen predicha se consideró un bloque de 2×2 como kernel para predecir el valor \hat{x} de cada pixel x , donde los valores del bloque son las intensidades de los vecinos inmediatamente horizontales a , verticales b y diagonales c del pixel a predecir. Éste fue descrito en [24] y se considera un detector de bordes primitivo; su arreglo por bloque se muestra en la Fig. 6.2.

La predicción \hat{x} del valor del pixel x se obtiene mediante (6.1).

c	a
b	x

Figura 6.2: Arreglo por bloque del detector de bordes. Fuente: [Autor].

$$\dot{x} = \begin{cases} \max(a,b) & c \leq \min(a,b) \\ \min(a,b) & c \geq \max(a,b) \\ a + b - c & \text{otro} \end{cases} \quad (6.1)$$

Finalmente, al extraer las características LPB de 59-D de los canales rojo y verde de la imagen original, la imagen PE y la sub-banda horizontal de la transformada Wavelet, y concatenarlas el método entregó 354 componente en el dominio espacial las cuales son la representación o huella digital de cada cámara. Posteriormente, estas características se usaron para entrenar los clasificadores y así obtener la exactitud de la técnica.

6.2. Técnica basada en Patrones Binarios Locales y Cuantificación de Fase Local (LBP-LPQ)

Para esta se trabajó con el modelo de color HSV ya que al comparar los modelos de color RGB, HSV y YCrCb se encontró que las características derivadas de los componentes H y S de la imagen y su versión de ruido residual son altamente relevantes.

Primero, se procedió a llevar las imágenes del modelo RGB al HSV. Luego, se extrajeron características LBP de 59-D y LPQ de 59-D a los canales H y V de la imagen original y de la imagen de ruido residual (NI), adicionalmente, solo las características LBP de 59-D a las componentes de alta frecuencia de la transformada de contorno de estos canales. El diagrama de flujo del método se presenta en Fig. 6.3.

Primero se transformaron las imágenes al modelo de color HSV y luego se calculó la imagen de ruido residual (NI), la cual se obtuvo de la resta de la imagen original y la imagen sin ruido (6.2).

$$NI = I - f(I) \quad (6.2)$$

Donde I es la imagen original y $f(\cdot)$ es un filtro de eliminación de ruido descrito en el apéndice A de [21], el cual utilizó una varianza local $\sigma^2 = 3$ la cual fue mencionada en [2].

Seguidamente, a la imagen de ruido residual se le calculó la transformada de contorno [25]

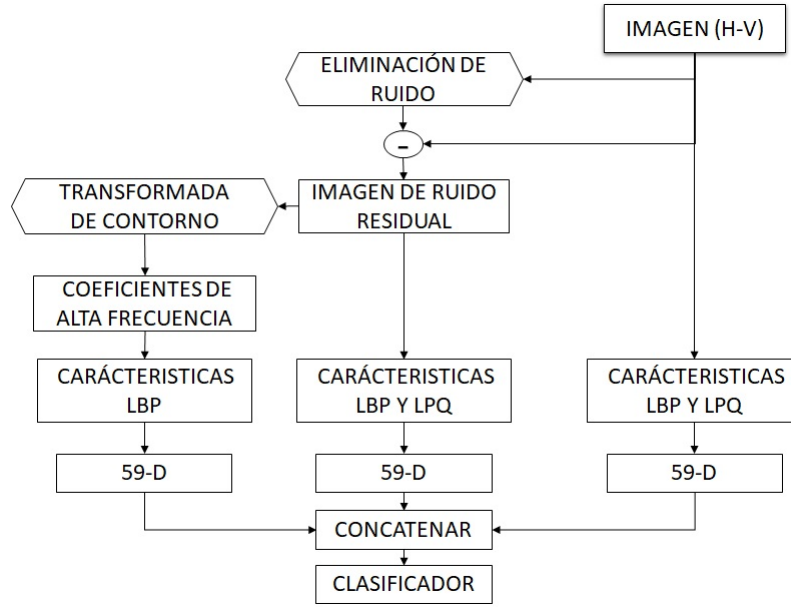


Figura 6.3: Diagrama de flujo de la técnica 2. Fuente: [Autor].

de dos niveles para luego, calcularle cuatro matrices de coeficientes de alta frecuencia mediante una transformada Wavelet Haar de cuarto nivel. A continuación, se extrajeron las características LBP 59-D de estas cuatro matrices y se denotaron como $F(k)_1^{lbp}, F(k)_2^{lbp}, F(k)_3^{lbp}, F(k)_4^{lbp}$ donde $k = H, V$. Finalmente, se obtuvieron las características LBP y LPQ de la imagen original I y la NI y se indicaron como $F(k)_5^{lbp}, F(k)_1^{lpq}, F(k)_6^{lbp}, F(k)_2^{lpq}$, respectivamente.

Por último, se concatenaron todas las características en un solo vector el cual es el vector propio representativo de la cámara (6.3).

$$F = \{F(k)_1^{lbp}, F(k)_2^{lbp}, F(k)_3^{lbp}, F(k)_4^{lbp}, F(k)_5^{lbp}, F(k)_6^{lbp}, F(k)_1^{lpq}, F(k)_2^{lpq}\} \quad (6.3)$$

El vector propio F se extrajo del dominio espacial y del dominio de la frecuencia de los canales de color de tono y valor, esta técnica al final de todo el proceso entregó 708 características LBP y 236 características de LPQ para un total de 944 características las cuales constituyeron la huella digital de la cámara.

Con el fin de cumplir el objetivo dos, el cual consistió en evaluar las anteriores técnicas, se tomaron dos instancias de evaluación las cuales se describen en los siguientes apartados.

6.3. Instancia 1: Evaluación con base de datos Dresden

Esta etapa consistió en probar la efectividad de los algoritmos usando una galería de fotografías tomadas con cámaras digitales compactas y réflex denominada Dresden Image Database [12] la cual contiene más de 14000 imágenes provenientes de alrededor de 73 cámaras y 25 modelos.

Para esto se descargaron 400 fotografías, 50 por cada cámara en formato JPEG recortadas desde el centro a un tamaño de 1200×1200 pixeles provenientes de 8 modelos diferentes (ver cuadro 6.2) de cámaras y 10 escenarios, 5 por cada escenario. Con las fotografías obtenidas se procedió a probar los algoritmos implementados y verificar su efectividad utilizando la herramienta de Matlab denominada Classification Learner (CL) [26] y con esta se obtuvo la matriz de confusión perteneciente a este experimento con las SVM disponibles. Se tomaron los datos obtenidos en el CL al validar los algoritmos y se compararon con los datos encontrados en la bibliografía para determinar si los resultados son los esperados, además, se validaron con otros clasificadores como Decision TREE, Discriminant Analysis y K-Nearest Neighbors (KNN) disponibles en la herramienta para comprobar si las SVM entregan la mejor exactitud. Al comprobar la efectividad de los algoritmos implementados se procedió a continuar con la segunda instancia.

Modelo de Cámara	Tipo	Abreviación
Canon Ixus 70	Compacta	Can_70
Casio EX-Z150	Compacta	Cas_Z1
Kodak M-1063	Compacta	Kod_M
Nikon Cool Pix-S7	Compacta	Nik_S7
Nikon D-70	Réflex	Nik_D7
Nikon D-200	Réflex	Nik_D2
Olympus mju-10	Compacta	Oly_m1
Samsung L-74	Compacta	Sam_L

Cuadro 6.2: Modelos de Cámara y su tipo.

6.4. Instancia 2: Evaluación con galería propia del grupo de investigación

Para esta segunda etapa se creó una galería de fotografías propia del grupo de investigación de Robótica Aplicada en la línea Visión por Computador y Aprendizaje de Máquina adscrito al mismo grupo, la cual contiene una cantidad de 660 fotografías tomadas con 33 cámaras de dispositivos móviles. Los 33 dispositivos provienen de 7 marcas y 27 modelos de celulares los cuales se presentan en el cuadro 6.3.

Para la creación de esta galería se utilizaron los 4 cuadros de fondo plano presenten en la Fig. 6.4, a los cuales se les tomaron 5 fotos en ráfaga sin ser rotados. Estas fueron tomadas utilizando un trípode con adaptador para celulares el cual estaba ubicado a 130cm de los cuadros como se observa en Fig. 6.5.



(a) Esquema UTP.



(b) Esquema Bus.



(c) Esquema Cartagena.



(d) Esquema bahía Hong Kong.

Figura 6.4: Esquemas elegidos para la creación de la galería propia del grupo. Fuente: [Autor].

Marca	Modelo	Abreviación	Marca	Modelo	Abreviación
Asus	Zenofone 4	Asu_Ze4	Motorola	Moto G6 play	Mot_G6ply
	Zenofone 4+5	Asu_Ze4+		Moto G6 plus	Mot_G6pls
Huawei	Mate 10 lite	Hua_M10L	Samsung	A6	Sam_A6
	P10 lite	Hua_P10L		A30	Sam_A3
	P20 lite	Hua_P20L		Edge s6	Sam_S6
	Y9	Hua_Y9		Galaxy J4	Sam_J4
Iphone	6	Iph_6		J2 prime	Sam_J2
	6 plus	Iph_6p		J6	Sam_J6
	7	Iph_7		J7 metal	Sam_J7m
LG	K8	Lg_K8		J7 neo	Sam_J7n
Motorola	Moto G2	Mot_G2		J7 prime	Sam_J7p
	Moto G4s plus	Mot_G4s		S5	Sam_S5
	Moto G5s plus	Mot_G5s	Xiaomy	Note 5	Xia_N5
	Moto G6	Mot_G6			

Cuadro 6.3: Marcas y modelos de celulares en galería propia.



Figura 6.5: Montaje de fotografías. Fuente: [Autor].

Para determinar la cantidad de celulares y modelos a utilizar en esta instancia se cruzó la información de las marcas más comunes y los modelos disponibles dentro de la base de datos creada. Con lo anterior, se eligieron las marcas Asus, Iphone, Motorola y Samsung para determinar la exactitud de las técnicas al discriminar marcas ya que estas cuentan con dos o más modelos disponibles,

a esta prueba se le conocerá como experimento 1.

Para determinar la exactitud de las técnicas al excluir modelos (experimento 2) se probó primero entre todos los modelos de la marca Huawei, Motorola y Samsung, por último se eligieron al azar 4 modelos de las marcas Motorola y Samsung y el resto de los modelos disponibles en 6.3, los dispositivos elegidos se presentan en el cuadro 6.4. A estas verificaciones se les nombró como experimento 2.1 (modelos marca Huawei), experimento 2.2 (modelos marca Motorola), experimento 2.3 (modelos marca Samsung) y experimento 2.4 respectivamente (modelos mezclados).

Marca	Modelo	Marca	Modelo
Asus	Zenofone 4	Motorola	Moto G2
	Zenofone 4+5		Moto G4s plus
Huawei	Mate 10 lite		Moto G6 play
	P10 lite		Moto G6 plus
	P20 lite	Samsung	A6
	Y9		Galaxy J4
Iphone	6		J7 metal
	6 plus		S5
	7	Xiaomy	Note 5
LG	K8		

Cuadro 6.4: Modelos elegidos para experimento 2.4

Para probar la exactitud de las técnicas al aislar o individualizar los dispositivos iguales entre modelos y marcas diferentes el cual se nombró experimento 3. Para este se eligieron los modelos presentes en el cuadro 6.5 ya que estos contaban con dos ejemplares cada uno. Luego se procedió a extraer las características representativas de las técnicas con los dispositivos elegidos para cada experimento y se midió el tiempo de ejecución de cada algoritmo al evaluar las fotografías.

Marca	Modelo
Asus	Zenofone 4
Huawei	Y9
Motorola	Moto G4s plus
Samsung	Galaxy J4
	J6
	J7 prime

Cuadro 6.5: Dispositivos para experimento 3.

Por último, se comprobó la exactitud y efectividad de los algoritmos al tratar fotografías tomadas con celulares, para esto se recortaron las imágenes a un tamaño de 1050×1050 desde el centro de la misma, se ejecutaron los algoritmos y se llevaron los datos extraídos de la galería creada al CL de Matlab. Para fijar la mejor exactitud se utilizaron las opciones posibles dentro de las familias

de clasificadores conocidas como: Decision TREE, Discriminant Analysis, KNN y SVM.

Finalmente, se procedió a la selección de la mejor técnica para cumplir así con el último objetivo del proyecto, para lo cual primero se tomaron los datos obtenidos en las dos instancias y se procedieron a tabular. Para la primera instancia se tabularon los resultados de exactitud entregados por el mejor clasificador probado, esto para lograr comparar los valores obtenidos en la literatura y los obtenidos en la propuesta actual para cada método. Para la segunda instancia se tabuló el clasificador, la exactitud, la precisión, la velocidad de predicción por imagen y el tiempo de entrenamiento entregados por el CL y se escogió el mejor clasificador el cual fue aquel que presentó mejores resultados teniendo en mayor consideración de la exactitud y la velocidad de predicción .

Capítulo 7

Resultados

Los resultados presentes en este capítulo se obtuvieron mediante un análisis multiclase y los datos indicados en los cuadros muestran los valores generales entregados por los diferentes modelos de entrenamiento.

7.1. Primera instancia: Evaluación con base de datos Dresden

Los resultados de exactitud, precisión, velocidad de entrenamiento y tiempo de respuesta arrojados al evaluar las técnicas con los clasificadores de las familias Decision Trees, Discriminant Analysis, SVM y KNN presentes en el CL para el experimento 1 con cámaras digitales se presentan en los cuadros 7.1 (método LBP) y 7.2 (método LBP-LPQ). Estos valores se hallaron mediante una validación cruzada con 5 iteraciones de los datos proporcionados por las técnicas.

Clasificador		Resultados			
Familia	Kernel	Exactitud General (%)	Precisión General (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	86,8	86,0	3800	2,49
	Medium	86,8	86,0	5100	0,58
	Coarse	57,0	57,0	4900	0,54
Discriminant Analysis	Linear	95,5	95,0	2200	10,12
SVM	Linear	97,3	97,0	1700	7,44
	Cubic	98,8	99,0	1600	2,30
	Medium Gaussian	98,0	98,0	1500	2,28
KNN	Fine	97,3	97,0	3300	1,46
	Cubic	97,0	97,0	680	1,96
	Weighted	97,3	97,0	4000	0,46

Cuadro 7.1: Resultados generales para el instancia 1 método LBP.

De 7.1 se observó que el clasificador con mayor exactitud para la técnica LBP en este experimento es un SVM con kernel cúbico, del cual se presenta la matriz de confusión en Fig. 7.1. Por otro

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	87,3	87,0	1200	3,88
	Medium	87,3	87,0	1600	1,48
	Coarse	55,3	55,0	1800	1,28
Discriminant Analysis	Linear	98,8	99,0	1200	2,35
SVM	Linear	97,8	98,0	850	5,17
	Cubic	98,8	99,0	700	4,08
	Medium Gaussian	97,8	98,0	670	4,05
KNN	Fine	96,0	96,0	1100	1,94
	Cubic	87,5	88,0	270	4,99
	Weighted	92,3	92,0	1000	1,34

Cuadro 7.2: Resultados generales para el instancia 1 método LBP-LPQ.

lado, de 7.2 se determinó que el mejor clasificador para la técnica LBP-LPQ fue un Discriminant Analysis con kernel lineal, su matriz de confusión se presenta en Fig. 7.2.

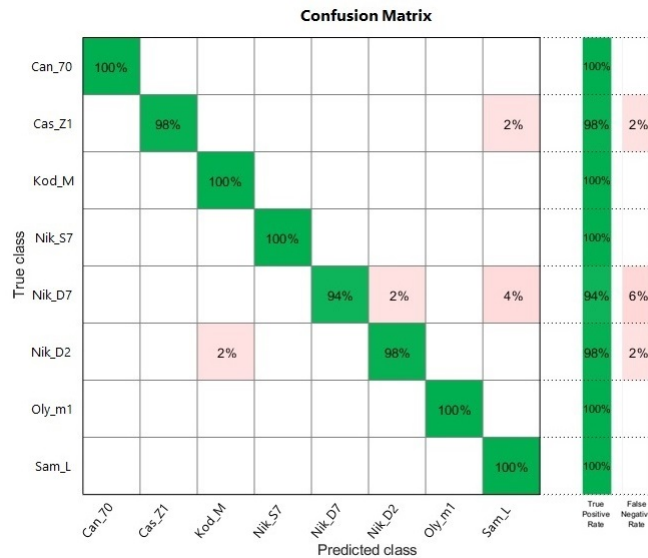


Figura 7.1: Matriz de confusión método LBP de SVM Cubic para instancia 1.

Al comparar los resultados entregados por el mejor clasificador para la propuesta actual y los obtenidos por Xu en [11] se logró observar que Xu presentó una mayor exactitud con las cámaras Casio Ixus 70 y la Nikon D-70, pero menor con la Nikon D-200 (ver 7.3).

Al inspeccionar los valores de exactitud obtenidos por Wang en [13] y compararlos con los obtenidos por medio del mejor clasificador para la propuesta actual (Discriminant Analysis con kernel lineal Fig. 7.2) se encontró que para las cámaras en común se presentan iguales exactitudes a excepción de la cámara Olympus Mju la cual es menor que la obtenida por Wang (cuadro 7.4).

Cámara	Exactitud (%)	
	Xu [11]	Propuesta Actual
Canon Ixus 70	100,0	100,0
Casio Ex-Z150	99,5	98,0
Kodak M-1063	100,0	100,0
Nikon Cool Pix-S7	100,0	100,0
Nikon D-70	100,0	94,0
Nikon D-200	97,6	98,0
Olympus Mju	100,0	100,0
Samsung L-74	100,0	100,0

Cuadro 7.3: Comparación de resultados generales obtenidos por Xu y la propuesta actual para instancia 1.

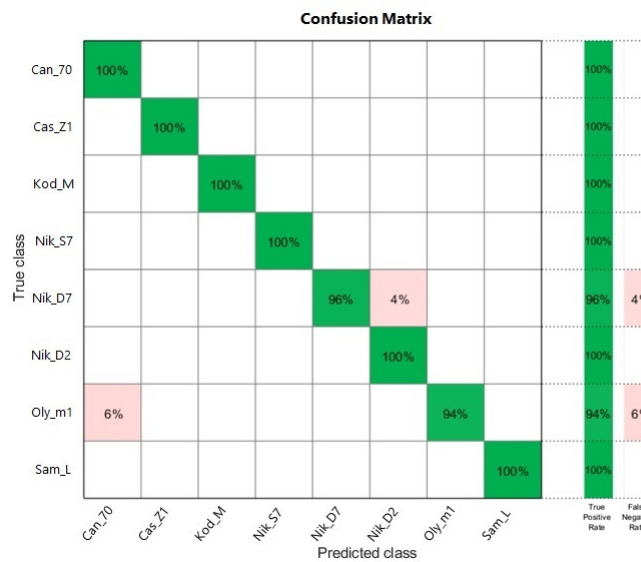


Figura 7.2: Matriz de confusión del clasificador Discriminant Analysis Linear para instancia 1 método LBP-LBP.

7.2. Segunda instancia: Evaluación con galería propia del grupo de investigación

7.2.1. Resultados para el método LBP

Los resultados obtenidos con los diferentes clasificadores para el experimento 1 de la segunda instancia de la evaluación (marcas de celulares), para el método LBP, se presentan en la tabla 7.5 donde se observó que el mejor clasificador fue el KNN con kernel ponderado (Weighted) con una exactitud del 100 %; su matriz de confusión se presenta en Fig. 7.3.

Cámara	Exactitud (%)	
	Wang et al [13]	Propuesta Actual
Canon Ixus 70	100,0	100,0
Casio Ex-Z150	100,0	100,0
Kodak M-1063	100,0	100,0
Olympus Mju	100,0	100,0
Samsung L-74	100,0	94,0

Cuadro 7.4: Comparación de resultados generales obtenidos por Wang y la propuesta actual.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	90,5	91,0	1400	0,96
	Medium	90,5	91,0	1400	0,79
	Coarse	85,0	85,0	1400	0,75
Discriminant Analysis	Linear	100,0	100,0	1100	1,09
SVM	Linear	100,0	100,0	800	3,70
	Cubic	99,5	99,0	830	1,78
	Medium Gaussian	100,0	100,0	750	1,80
KNN	Fine	100,0	100,0	1000	1,21
	Cubic	96,5	97,0	690	1,12
	Weighted	100,0	100,0	1200	0,67

Cuadro 7.5: Resultados generales obtenidos para experimento 1 método LBP.

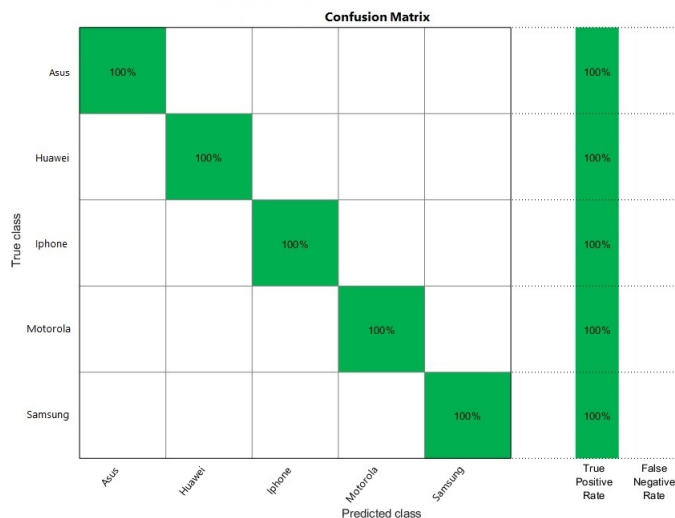


Figura 7.3: Matriz de confusión del clasificador KNN Weighted para Experimento 1 instancia 2 método LBP.

Los resultados para la discriminación entre los modelos de la marca Huawei (experimento 2.1),

los modelos de la marca Motorola (experimento 2.2), los modelos de la marca Samsung (experimento 2.3) y los modelos de diferentes marcas (experimento 2.4) con el método LBP se presentan en los cuadros 7.6, 7.7, 7.8, 7.9 respectivamente.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	97,5	98,0	540	0,85
	Medium	97,5	98,0	510	0,70
	Coarse	97,5	98,0	530	0,72
Discriminant Analysis	Linear	100,0	100,0	510	0,92
SVM	Linear	100,0	100,0	420	1,47
	Cubic	100,0	100,0	400	1,29
	Medium Gaussian	100,0	100,0	390	1,24
KNN	Fine	100,0	100,0	500	0,90
	Cubic	85,0	85,0	380	0,77
	Weighted	100,0	100,0	470	0,74

Cuadro 7.6: Resultados generales obtenidos para experimento 2.1 método LBP.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	95,0	95,0	810	0,98
	Medium	95,0	95,0	790	0,78
	Coarse	79,2	79,0	800	0,69
Discriminant Analysis	Linear	100,0	100,0	680	1,07
SVM	Linear	100,0	100,0	500	2,33
	Cubic	100,0	100,0	480	2,09
	Medium Gaussian	100,0	100,0	460	1,97
KNN	Fine	99,2	99,0	700	0,96
	Cubic	93,2	93,0	520	0,87
	Weighted	99,2	99,0	750	0,61

Cuadro 7.7: Resultados generales obtenidos para experimento 2.2 método LBP.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	90,0	90,0	1000	3,94
	Medium	90,0	90,0	1500	0,81
	Coarse	47,5	47,0	1400	0,74
Discriminant Analysis	Linear	98,0	98,0	990	1,93
SVM	Linear	98,9	99,0	430	7,10
	Cubic	99,5	99,0	480	4,94
	Medium Gaussian	98,0	98,0	430	4,59
KNN	Fine	99,0	99,0	920	1,42
	Cubic	91,0	91,0	570	1,28
	Weighted	95,0	95,0	1200	0,68

Cuadro 7.8: Resultados generales obtenidos para experimento 2.3 método LBP.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	92,1	92,0	1800	4,26
	Medium	92,1	92,0	2300	1,72
	Coarse	25,8	26,0	2300	1,19
Discriminant Analysis	Linear	98,7	99,0	1800	1,89
SVM	Linear	99,7	99,0	280	19,00
	Cubic	99,7	99,0	300	15,97
	Medium Gaussian	100,0	100,0	280	17,17
KNN	Fine	100,0	100,0	1700	1,52
	Cubic	92,1	92,0	640	2,09
	Weighted	100,0	100,0	2000	0,80

Cuadro 7.9: Resultados generales obtenidos para experimento 2.4 método LBP.

De los cuadros 7.6, 7.7, 7.8 y 7.9 se determinó que los mejores clasificadores para los experimentos 2.1 y 2.2 fue un Discriminant Analysis lineal, para el 2.3 y 2.4 fueron SVM Cubic y KNN Weighted respectivamente. En las Fig. 7.10, 7.11, 7.12 y 7.13 se presentan las matrices de confusión de los clasificadores ya mencionados.

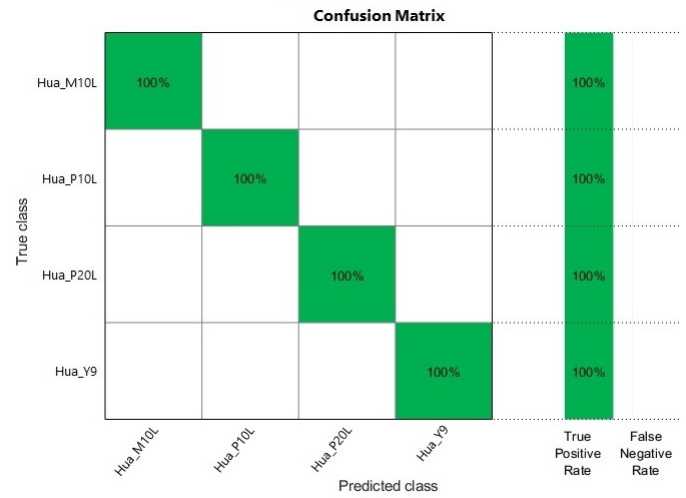


Figura 7.4: Matriz de confusión del clasificador Discriminant Analysis linear para el experimento 2.1.

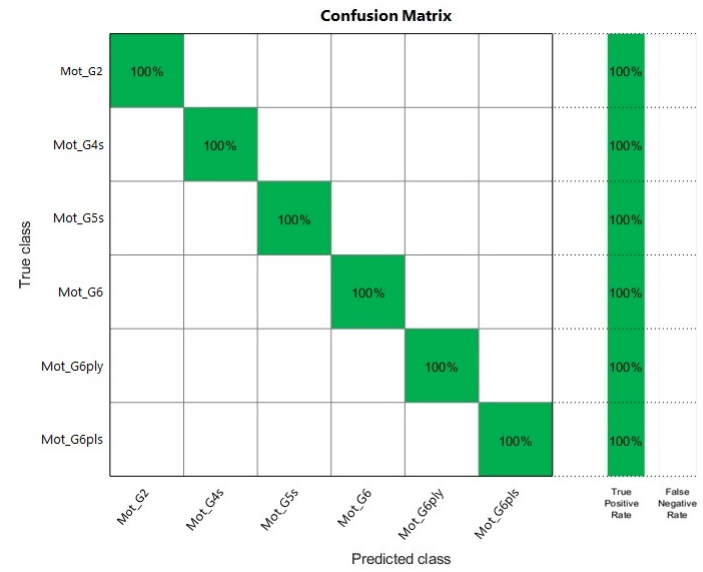


Figura 7.5: Matriz de confusión del clasificador Discriminant Analysis linear para el experimento 2.2.

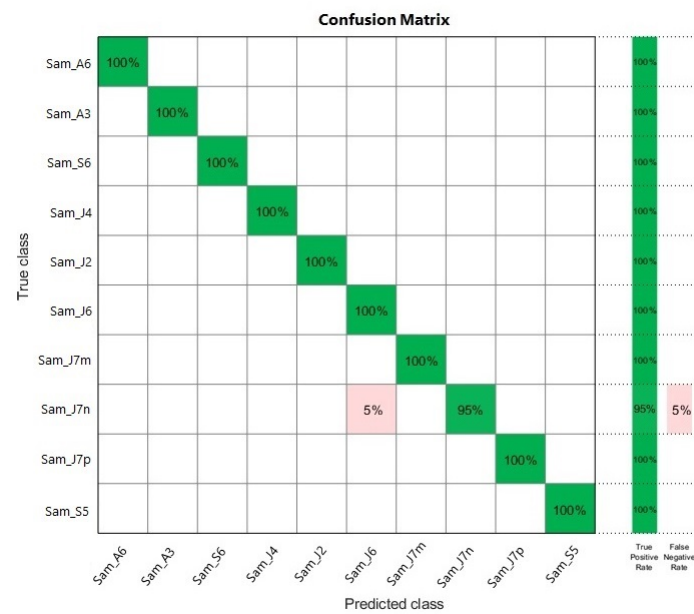


Figura 7.6: Matriz de confusión del clasificador SVM Cubic para el experimento 2.3.

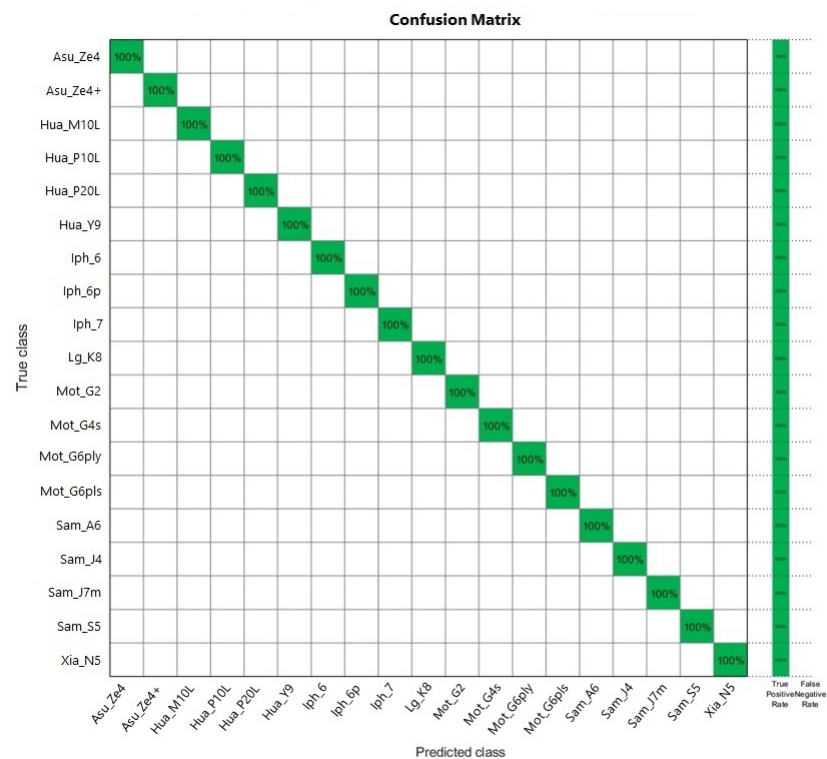


Figura 7.7: Matriz de confusión del clasificador KNN Weighted para el experimento 2.4.

Gracias a la Fig. 7.6 se pudo conocer que el método LBP para la detección de modelos entre la

marca Samsung de la base de datos creada presenta una confusión del 5 % entre el modelo J7 neo y J6 de la misma.

Los resultados obtenidos para el experimento 3 con el método LBP se presentan en el cuadro 7.10, de donde se consiguió detectar que el mejor clasificador para la técnica LBP al individualizar dispositivos fue una SVM con kernel Cubic.

Clasificador		Resultados			
Familia	Kernel	Exactitud General (%)	Precisión General (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	86,3	86,0	1100	3,97
	Medium	86,3	86,0	1600	1,12
	Coarse	40,0	40,0	1400	0,95
Discriminant Analysis	Linear	87,1	87,0	1100	1,85
SVM	Linear	92,1	92,0	410	9,27
	Cubic	95,8	96,0	390	6,95
	Medium Gaussian	89,2	89,0	350	7,06
KNN	Fine	95,4	95,0	1100	1,32
	Cubic	73,3	73,0	670	1,33
	Weighted	88,3	88,0	1300	0,72

Cuadro 7.10: Resultados generales obtenidos para experimento 3 método LBP.

De la matriz de confusión de la SVM Cubic (Fig. 7.8) se consiguió determinar que el algoritmo LBP presenta un error del 20% al individualizar el dispositivo Samsung Galaxy J4_1 ya que lo confunde con el dispositivo J7 prime_1 de la misma marca, y presenta confusión entre los dos dispositivos J6 de la misma.

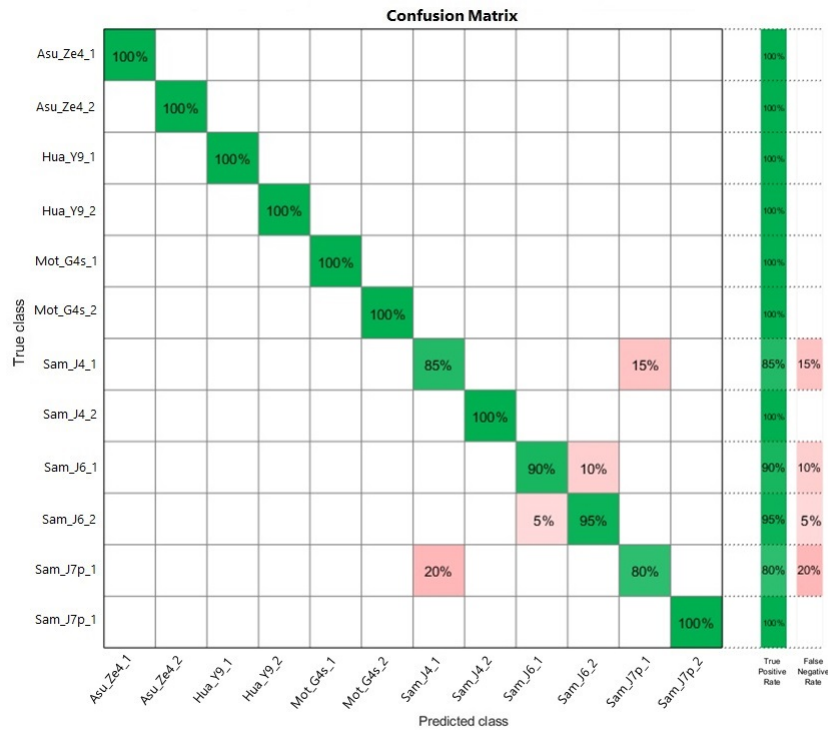


Figura 7.8: Matriz de confusión del clasificador SVM Cubic para Experimento 3 instancia 2 método LBP.

7.2.2. Resultados para el método LBP-LPQ

Los resultados obtenidos en el experimento 1 con el método LBP-LPQ (cuadro 7.11) permitieron determinar que el clasificador KNN Fine es la mejor opción para separar una marca de celular de otra y su matriz de confusión (Fig. 7.9) reafirma este hecho.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	93,0	93,0	360	5,19
	Medium	93,0	93,0	420	2,14
	Coarse	74,0	74,0	430	1,98
Discriminant Analysis	Linear	99,5	99,0	320	3,78
SVM	Linear	99,5	99,0	300	5,66
	Cubic	99,5	99,0	270	3,76
	Medium Gaussian	100,0	100,0	290	3,58
KNN	Fine	100,0	100,0	330	2,61
	Cubic	98,0	98,0	210	3,16
	Weighted	99,0	99,0	370	1,98

Cuadro 7.11: Resultados generales obtenidos para experimento 1 método LBP y LPQ.

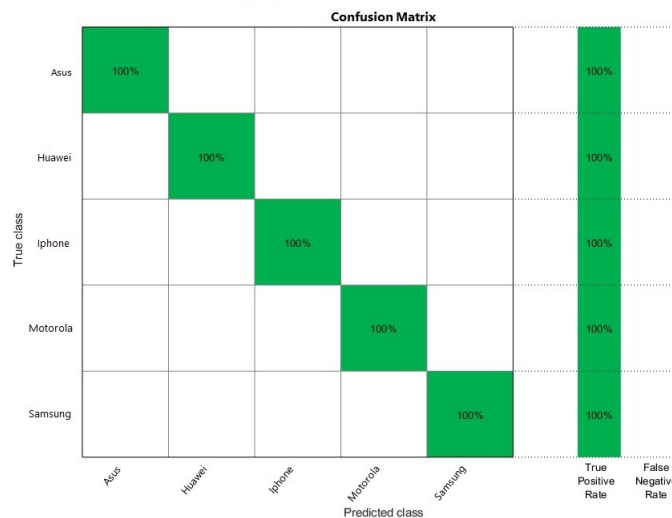


Figura 7.9: Matriz de confusión del clasificador KNN Fine para instancia 1 método LBP-LPQ.

Los resultados presentes en las tablas 7.12, 7.13, 7.14 y 7.15 representan los valores obtenidos al ensayar la técnica en los experimentos 2.1, 2.2, 2.3 y 2.4, respectivamente.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	95,0	95,0	130	3,78
	Medium	95,0	95,0	150	1,97
	Coarse	95,0	95,0	160	1,85
Discriminant Analysis	Linear	100,0	100,0	140	3,08
SVM	Linear	100,0	100,0	130	3,69
	Cubic	100,0	100,0	120	2,95
	Medium Gaussian	100,0	100,0	120	2,96
KNN	Fine	100,0	100,0	150	2,37
	Cubic	80,0	80,0	140	1,97
	Weighted	88,8	89,0	170	1,66

Cuadro 7.12: Resultados generales obtenidos para experimento 2.1 método LBP-LPQ.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	91,7	92,0	230	2,34
	Medium	91,7	92,0	230	2,18
	Coarse	80,0	80,0	220	2,18
Discriminant Analysis	Linear	100,0	100,0	200	2,35
SVM	Linear	100,0	100,0	200	3,84
	Cubic	100,0	100,0	170	3,77
	Medium Gaussian	100,0	100,0	170	3,64
KNN	Fine	99,2	99,0	220	2,13
	Cubic	85,0	85,0	160	2,44
	Weighted	96,7	97,0	230	1,88

Cuadro 7.13: Resultados generales obtenidos para experimento 2.2 método LBP-LPQ.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicación (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	68,5	68,0	400	3,38
	Medium	68,5	68,0	380	2,52
	Coarse	44,5	44,0	410	2,17
Discriminant Analysis	Linear	78,5	78,0	320	2,68
SVM	Linear	78,0	78,0	220	7,86
	Cubic	78,0	78,0	200	7,79
	Medium Gaussian	77,5	77,0	190	7,81
KNN	Fine	79,5	79,0	360	2,28
	Cubic	71,0	71,0	200	3,44
	Weighted	73,5	73,0	380	1,95

Cuadro 7.14: Resultados generales obtenidos para experimento 2.3 método LBP-LPQ.

Clasificador		Resultados			
Familia	Kernel	Exactitud (%)	Precisión (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	94,7	95,0	820	4,02
	Medium	94,7	95,0	820	3,76
	Coarse	25,8	26,0	820	2,52
Discriminant Analysis	Linear	100,0	100,0	650	3,12
SVM	Linear	98,9	99,0	200	24,46
	Cubic	99,2	99,0	160	25,42
	Medium Gaussian	100,0	100,0	140	25,47
KNN	Fine	100,0	100,0	710	2,29
	Cubic	92,9	93,0	220	5,48
	Weighted	96,6	97,0	690	2,04

Cuadro 7.15: Resultados generales obtenidos para experimento 2.4 método LBP-LPQ.

De las 4 tablas anteriores se declaró que los mejores clasificadores bajo los parámetros de mejor exactitud, precisión, velocidad de predicción y tiempo de entrenamiento son un KNN Fine para el experimento 2.1, 2.3, 2.4, y un Discriminant Analysis con kernel lineal para el experimento 2.2. Sus matrices de confusión se presentan en las Fig. 7.10, 7.11, 7.12 y 7.13.

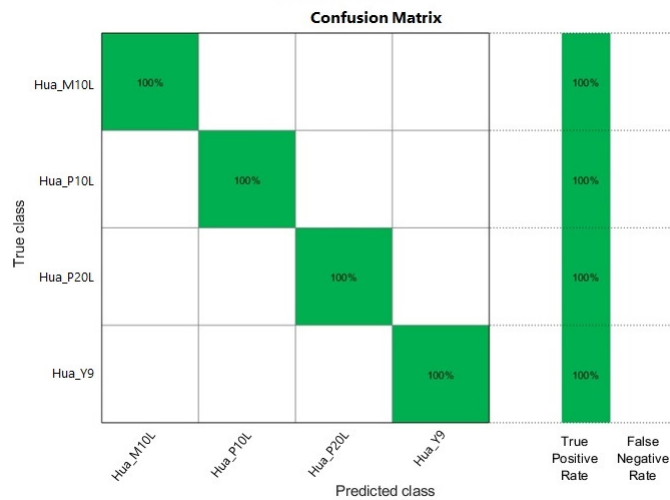


Figura 7.10: Matriz de confusión del clasificador KNN Fine para el experimento 2.1.

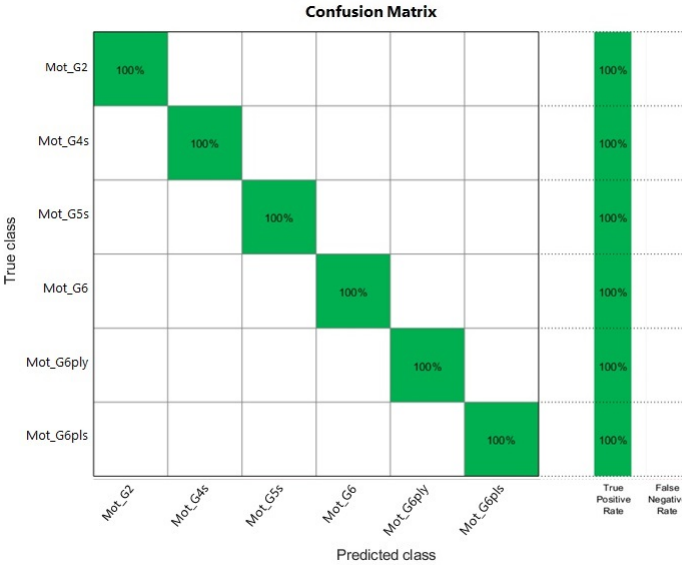


Figura 7.11: Matriz de confusión del clasificador Discriminant Analysis linear para el experimento 2.2.

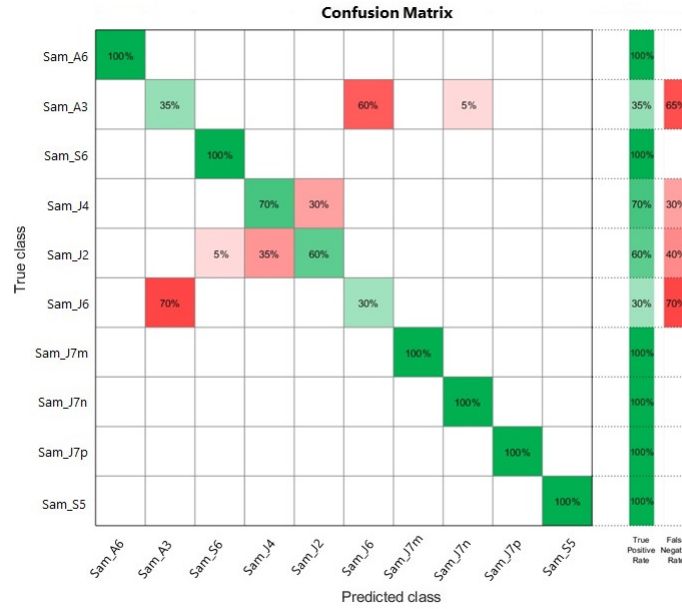


Figura 7.12: Matriz de confusión del clasificador KNN Fine para el experimento 2.3.

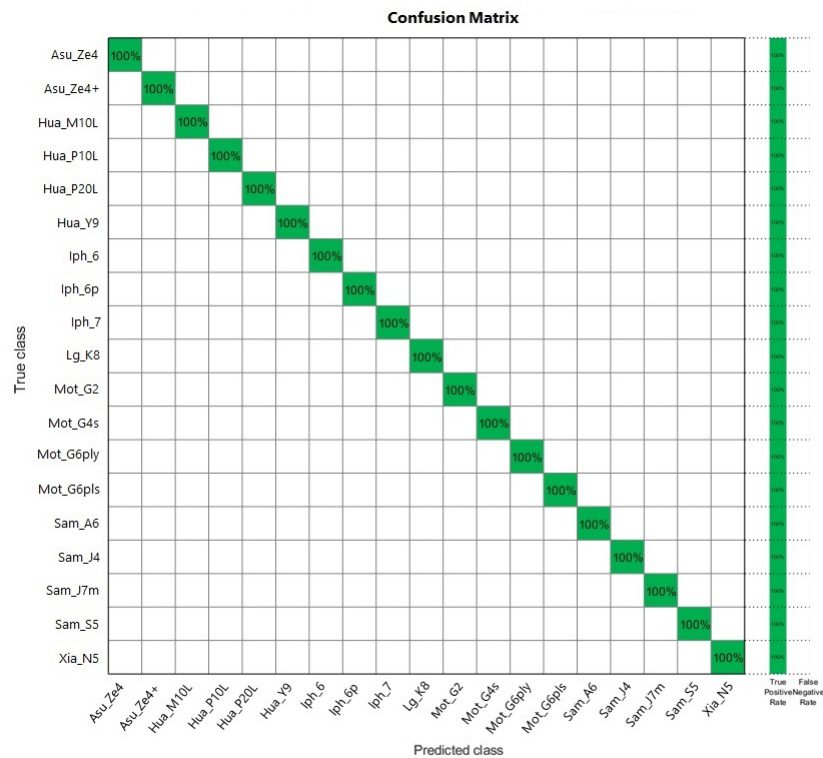


Figura 7.13: Matriz de confusión del clasificador KNN Fine para el experimento 2.4.

De 7.12 se observó que el algoritmo LBP-LPQ presenta una tasa de falsos positivos alto (entre el 60% y 70%) entre el modelo A30 y J6 de la marca Samsung. También, presenta un nivel de confusión más bajo (del 30% al 35%) entre los modelos Galaxy J4 y J2 prime de la misma marca, pero no presenta ninguna confusión entre los otros modelos. Los modelos nombrados anteriormente cuentan con características diferentes en sus cámaras, lo que lleva a pensar que los niveles de confusión surgen por problemas de la técnica.

Por ultimo, el mejor clasificador dentro del cuadro 7.16 para esta técnica al desarrollar el experimento 3 fue una SVM con kernel cuadrático, no obstante, este clasificador presenta una error del 100% al distinguir el segundo dispositivo del modelo Galaxy J4 ya que lo confunde con el segundo dispositivo del modelo J6 de Samsung, y los primeros dispositivos de los modelos J4 y J7 prime de la misma marca (ver Fig. 7.14).

Clasificador		Resultados			
Familia	Kernel	Exactitud promedio (%)	Precisión (%)	Velocidad de Predicción (Obs/s)	Tiempo de Entrenamiento (s)
Decision Trees	Fine	74,2	74,0	510	2,93
	Medium	74,2	74,0	520	2,64
	Coarse	40,8	41,0	520	2,22
Discriminant Analysis	Linear	73,8	74,0	440	2,56
SVM	Linear	73,3	73,0	250	10,15
	Quadratic	79,2	79,0	200	10,22
	Medium Gaussian	74,2	74,0	190	10,24
KNN	Fine	76,7	78,0	470	2,20
	Cubic	67,1	67,0	230	3,48
	Weighted	66,7	67,0	440	1,96

Cuadro 7.16: Resultados generales obtenidos para experimento 3 método LBP-LPQ.

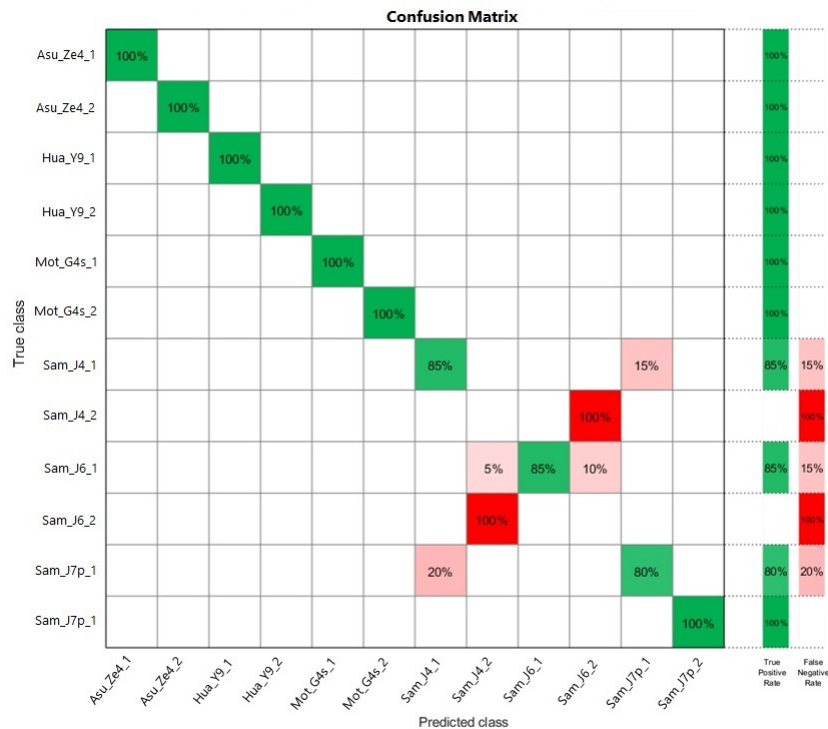


Figura 7.14: Matriz de confusión del clasificador SVM Quadratic para Experimento 3 método LBP-LPQ.

Los modelos presentes en el experimento 3 cuentan características similares en cuanto a los megapíxeles y la apertura del lente de sus cámaras por lo cual, no se logro encontrar una razón para que el modelo de entrenamiento del método LBP-LPQ presenta los niveles de confusión ilustrados en la Fig. 7.14.

Capítulo 8

Conclusiones

Se observó que al probar las técnicas LBP y LBP-LPQ con la base de datos Desden Image Database la mejor para discriminar modelos de cámaras digitales compactas o reflex fue la técnica LBP, ya que presentó la misma exactitud que la técnica LBP-LPQ, pero por su clasificador mostró una diferencia de 1,78s en el tiempo de entrenamiento y 900 observaciones por segundo más.

Los métodos LBP y LBP-LPQ presentaron la misma exactitud para los experimentos de discriminación de marcas de celulares (experimento 1), modelos dentro de la marca Huawei (experimento 2.1), modelos dentro de la marca Motorola (experimento 2.2) y discriminación de modelos de diferentes marcas (experimento 2.4) realizados con la base de datos creada, donde el método LBP entrego valores mayores de velocidad de predicción y menor tiempo de entrenamiento; esto se debe a que este método tiene una carga computacional mucho menor, ya que entrega menos características representativas por imagen.

El método LBP presentó una mejor exactitud al discriminar los modelos dentro de la marca Samsung (experimento 2.3) y en la individualización de modelos (experimento 3) dentro de la base de datos creada, presentando problemas dentro de los dispositivos de la marca Samsung. No se logro determinar una causa definitiva respecto a las tasas de falsos negativos en estos experimentos ya que, los modelos de celulares utilizados en estos presentan características similares en sus cámaras y los errores solo se presentan en algunos dispositivos.

Con los experimentos realizados en este trabajo con fotografías provenientes de cámaras compactas, reflex y de celulares de las bases de datos de Desdren y la base de datos creada por el grupo de investigación se logra concluir que el mejor método implementado para identificación de cámara fuente a partir de características de textura en imágenes digitales es el método LBP ya que presento mejores resultados de exactitud, precisión, velocidad de predicción y tiempo de entrenamiento.

Bibliografía

- [1] mobile operators worldwide, “annual report,” February 2019., Aug. 2019, it’s taken only the graphic of the report. [Online]. Available: <https://www.gsma.com/r/mobileeconomy/> 2
- [2] M. Chen, J. Fridrich, and M. Goljan, “Digital Imaging Sensor Identification Further Study,” in *Security, steganography, and watermarking of multimedia contents IX*, 2007. [Online]. Available: https://www.researchgate.net/publication/228590993_Digital_imaging_sensor_identification_further_study_-_art_no_65050P 6.2
- [3] X. K. Bo Wang and X. You, “Source camera identification using support vector machines,” in *Advances in Digital Forensics V*, ser. IFIP AICT, G. Peterson and S. Shenoi, Eds., vol. 306. Springer, 2009, p. 107â118. [Online]. Available: https://link-springer-com.ezproxy.utp.edu.co/chapter/10.1007/978-3-642-04155-6_8
- [4] Y. Long and Y. Huang, “Image based source camera identification using demosaicking,” in *2006 IEEE Workshop on Multimedia Signal Processing*. IEEE, 2006, pp. 419–424.
- [5] J. S. Ho, O. C. Au, J. Zhou, and Y. Guo, “Inter-channel demosaicking traces for digital image forensics,” in *2010 IEEE International Conference on Multimedia and Expo*. IEEE, 2010, pp. 1475–1480.
- [6] M. Kharrazi, H. Sencar, and N. Memon, “Blind source camera identification,” in *Proceedings of the 2005 International Conference on Image Processing*, 2005, pp. 69–72.
- [7] A. T. H. Ainuddin Wahid Abdul Wahab1 and S. Li, “Inter-camera model image source identification with conditional probability features,” 2012. 4
- [8] T. Ojala and M. Pietikainen, “Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns,” in *Computer Vision - ECCV 2000*. IEEE, 2002. 5.2, 6
- [9] V. Ojansivu and J. Heikkila, “Blur Insensitive Texture Classification Using Local Phase Quantization,” in *ICISP 2008*, ser. LNCS, A. Elmoataz *et al.*, Eds., vol. 5099. Springer, 2008, pp. 236–243. 5.3, 6

- [10] B. Wang, Y. Guo, X. Kong, and F. Meng, "Source camera identification forensics based on wavelet features," 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/5337441> 4
- [11] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in *2012 IEEE International Conference on Multimedia and Expo.* IEEE, 2012, pp. 392–397. 4, 6, ??, 6.1, 7.1, ??
- [12] T. Gloe and R. Bohme, "The Dresden Image Database for Benchmarking Digital Image Forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, 2010. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1774427> 4, 6.3
- [13] B. Xu, X. Wang, X. Zhou, J. Xi, and S. Wang, "Source camera identification from image texture features," *Neurocomputing*, vol. 207, pp. 131–140, 2016. 4, 6, ??, 7.1, ??
- [14] A. Rashidi, Farbod, and Razzazi, "Single image camera identification using i-vectors," in *7th International Conference on Computer and Knowledge Engineering (ICCCKE 2017), October 26-27 2017, Ferdowsi University of Mashhad.* IEEE, 2017. [Online]. Available: <https://ieeexplore-ieee-org.ezproxy.utp.edu.co/document/8167913> 4, 6
- [15] B. Wang, J. Yin, S. Tan, Y. Li, and M. Li, "Source camera model identification based on convolutional neural networks with local binary patterns coding," *Signal Processing: Image Communication*, vol. 68, pp. 162–168, oct 2018. [Online]. Available: <https://www-sciencedirect-com.ezproxy.utp.edu.co/science/article/pii/S0923596518307628> 4, 6
- [16] S. Gao, G. Xu, and R.-M. Hu, "Camera Model Identification Based on the Characteristic of CFA and Interpolation," in *IWDW 2011*, ser. LNCS, Y. Shi, H. Kim, and F. Perez-Gonzalez, Eds., vol. 7128. Springer, 2012, pp. 268–280. [Online]. Available: https://link-springer-com.ezproxy.utp.edu.co/chapter/10.1007/978-3-642-32205-1_22 5.1, 6
- [17] B. Sevinc, S. H. T., and M. Nasir, "Classification of digital camera-models based on demosaicing artifacts," *digital investigation*, vol. 5, no. 1-2, pp. 49–59, 2008. 5.1
- [18] J. R. Corripio, "Algoritmo de identificacion de fuente en imagenes and digitales," Master's thesis, Universidad Complutense de Madrid, 2013. 5.1, 5.4, 5.4
- [19] L. Serpa, "Diseno de un sistema inteligente de prediccion y simulacion para la prediccion y simulacion para la escritura en ninos con espasticidad," mathesis, Universidad Politecnica Saleciana, 2017. 5.4
- [20] M. Liu, N. Yu, and W. Li, "Camera Model Identification for JPEG Images via Tensor Analysis," *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010. [Online]. Available: <https://ieeexplore-ieee-org.ezproxy.utp.edu.co/document/5636047> 6

- [21] J. Luka, J. Fridrich, and M. Goljan, “Digital Camera Identification From Sensor Pattern Noise,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, jun 2006. 6, 6.2
- [22] A. R. Soobhany, A. Sheikh-Akbary, and Z. C. Schreuders, “Source camera identification using non-decimated wavelet transform,” in *Global Security, Safety and Sustainability - The Security Challenges of the Connected World*. Springer International Publishing, 2016, pp. 125–133. [Online]. Available: https://link-springer-com.ezproxy.utp.edu.co/chapter/10.1007/978-3-319-51064-4_11 6
- [23] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, “Source camera identification based on CFA interpolation.” IEEE, 2005. [Online]. Available: <https://ieeexplore-ieee-org.ezproxy.utp.edu.co/document/1530330> 6.1
- [24] M. J. Weinberger, G. Seroussi, and G. Sapiro, “Loco-i: A low complexity, context-based, lossless image compression algorithm,” in *I40*. IEEE, 1068. 6.1
- [25] D. Minh and V. Martin, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Transactions on image processing*, vol. 14, no. 12, pp. 2091–2106, 2005. 6.2
- [26] Mathworks, “Modelos de clasificacion de trenes en la aplicacion clasificacion aprendiz- matlab & simulink- mathworks america latina,” La.mathworks.com, Nov. 2019. [Online]. Available: <https://la.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html#bu3xete> 6.3